Optical Music Recognition

Antonio José Moya Díaz Fernando Pérez Bueno

ABSTRACT.- This document presents the steps and the decisions taken in the implementation of this project of optical music recognition. There are three main steps. Firstly, every element needs to be isolated. Then the element is recognized to identify which element is. Finally, in case of being a note, the value is calculated depending on its position on the score.

1 Introduction

The work presented in this document was the final project for the course *Digital Image Processing*. The aim of this project was to learn and train certain skills in the field of image processing instead of creating a complete music recognition software. That is why this project presents several limitations.

In the following sections the whole process is explained in detail, these limitations are pointed out and, in most cases, solutions for these limitations are given for possible future versions of this project.

2 Segmentation

It is essential for reading the score to focus on each of its elements one by one. Each element needs to be isolated in a single image in order to proceed with its identification.

The first step to achieve this is to divide the image horizontally using the function *corte_horizontal*. Thus, every horizontal element will be divided in a separate image (figure 1). These elements will be the staff and also every add-on that the image may contain such us title, author, lyrics, etc.

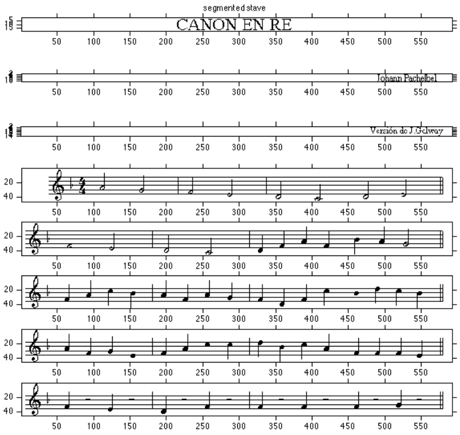

Figure 1: Output of the function corte horizontal

Then, the function *recon_pentagramas* is applied to the images obtained in the previous step. This function saves the images which contain staves while dismissing the other images. Staves are identified calculating histograms for every image and selecting those ones which have a stave-like histogram. Those histograms contain five maximums, one for each line of the stave.

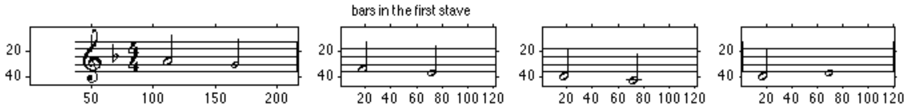

Figure 2: Output of the function separar compases

The second step is to divide the staves into bars. This task is easy given that each bar-line presents a maximum in a vertical histogram of every single stave. However, some staves do not have a bar-line at the beginning neither at the end, so the requirements to divide need to be relaxed and then filter the divided segments to recognize which ones are bars. This process is made by the function *separar_compases* (figure 2).

The final step of the segmentation process is to isolate any element inside every bar. This is made by the function *separar_elementos* which uses histograms again (figure 3).

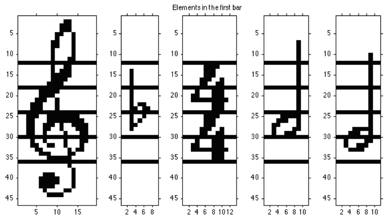

Figure 3: Output of the function separar elementos

The bar splitting is not truly needed given that this software can detect every element directly. However, this bar segmentation throws data which could be used in future or complex versions of this project.

3 Element Identification

At this point every element found in the stave is in a separate image. The next step is to identify which element is. However, before the identification, the image needs to be cleaned up in order to erase the stave lines. Finally the image is compared through correlation with the elements stored in the database. This way name and type of the element will be determined according to the elements in the database.

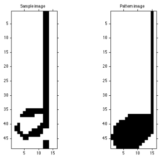

Figure 4: Comparison between the sample image to recognize (left) and a pattern image (right)

Database

The database used in this project is a very basic one, which contains only the elements needed to identify the scores given. Nevertheless, it was done in a scaleable way so the addition of new elements is easy. The database is organized in 3 tables: one for notes, one for rests and one for other elements.

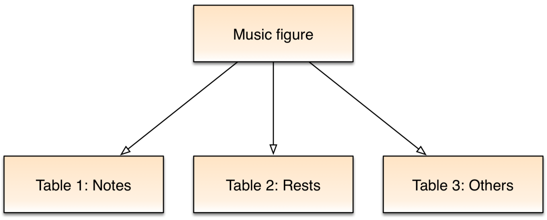

Figure 5: Database hierarchy

In case of identifying a note, the information of its pitch needs to be calculated. In case of a rest, only the information of its time length is needed. In contrast, this software does not calculate the information of the elements in the third group (clefs, time signatures, etc) but, as the element is recognized, this information could be obtain in a future version of this project.

Besides the organization of the database, it is important to use pattern images properly selected to match with the elements to be identified. If the element is not correctly centered in the pattern image or it is not correctly sized the success rate of the comparison will decrease significantly.

4 Pitch

After recognize a note, it is necessary to determine its pitch. Using the horizontal histogram, the note head is isolated.

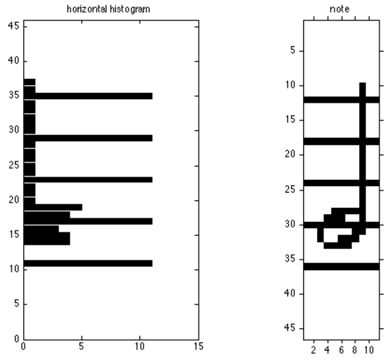

Figure 6: Note and its horizontal histogram

Then the image is divided in as many horizontal areas as pitches can be measured. These areas are calculated dynamically according to the stave, its size, and the space between the lines. The pitch is calculated just checking in which area the note head is.

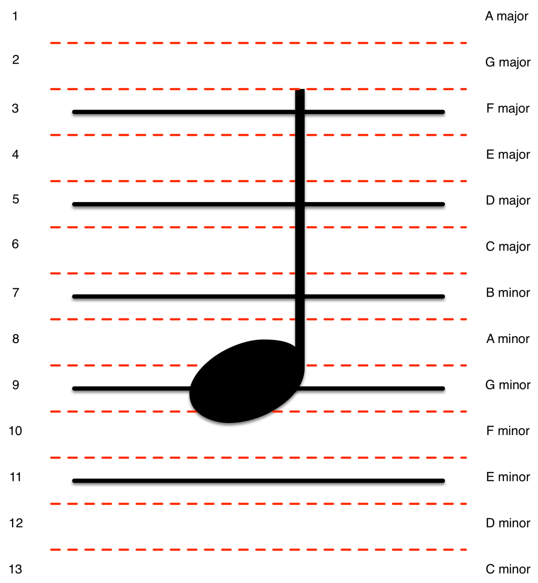

Figure 7: Space division scheme

Note that each position in the stave represents a pitch, but what pitch is depends on the Clef. This project calculates pitches considering it is always the G-Clef. However, this software is capable of recognizing the Clef so it would be easy, for future versions, to make the pitch calculation clef-sensitive.

The implemented procedure works only with simple notes, but it can be extended to recognize chords (more than one note at the same time). This could be done by checking the length of the head note positions (in case of two head notes were too close) or if more than one head note were detected (in case of two separated head notes).

5 Process

To sum up, for reading the score the first step is to isolate the staves, then the bars in the stave and, finally, every element in each bar. Then, after finishing stave, the software follows up with the next stave and so on up to finish the score.

Information extracted from each element is stored in a string variable in which newline is added after every stave in order to make a clearer display.

6 Conclusions

Although this is not a conceptually complex project, several problems were found during its development. Most of the notes are similar in shape, so the cleaning process needs to be very precise in order to correctly identify the image. In particular, whole and half rests are so similar that an extra checking was needed.

A fairly good quality pattern images were also needed. In fact, it seems that reduction in the size of the image to be identified decreases highly the success rate so that it is needed that the pattern image will always be bigger than the sample image.

Moreover, the development process should be done in the most generic way because trying to improve the recognition of a specific note value can damage others.

All these problems shows that the use this pattern recognition process alone is bit poor, and the whole process should be improve by using other techniques, for example, statistics based methods.